

Meteor observation with the SOURCE CubeSat – Developing a simulation to test on-board meteor detection algorithms

Marcel Liegibel¹, Jona Petr^{2,3}, Philipp Hoffmann², Niklas Geier², Sabine Klinkner³

Abstract

The scientific mission objectives of the Stuttgart Operated University Research CubeSat for Evaluation and Education are meteor observation, measurement of the lower Earth's atmosphere during re-entry as well as technology demonstrations. The meteor observation is done by pointing a camera towards Earth and continuously taking images during Eclipse. Since it is not possible to downlink all images, an on-board detection algorithm is necessary and mission critical. Therefore, this algorithm needs to be tested thoroughly. Realistic test data showing meteors from orbit is needed to properly develop and test the algorithm. Existing videos, provided by the Planetary Exploration Research Center, captured from the ISS are used as a baseline but are not sufficient to test the algorithm. The videos do not have the diversity of meteors needed and the meteor properties are not settable which makes it difficult to test the detection algorithm in as many scenarios as possible. Therefore, an artificial meteor program was developed to simulate meteors with given properties as perceived from a meteor observation system in a low Earth orbit. Here, we present the details of the artificial meteor program, its working principle and how we tested an algorithm for meteor detection.

The user can choose between different background videos, the existing ISS videos from PERC or the self-generated videos. Each different background is used to test a different aspect of the meteor detection algorithm. The ISS videos from PERC provide more diverse backgrounds than the self-generated videos with e.g., clouds and lightning. For these self-generated videos, a program is developed to take image sections of NASA's Black Marble and putting them frame by frame together into a video. These videos are more suitable for simulating satellite rotation and camera properties.

Independent of the background video, settable meteor properties contain important characteristics of a meteor like the light curve, brightness, speed, direction and shape. Additionally, the user can choose the meteor position in the video frame, in which frame it appears and which distance it covers. Furthermore, distortion settings can be applied which contain airplanes with adjustable parameters and scalable noise.

Only a properly working meteor detection algorithm leads to a success of a mission critical part of the SOURCE CubeSat. Therefore, the development of this artificial meteor generation program is crucial. Furthermore, this technology demonstration of developing and especially testing a meteor detection algorithm will enable future space-based missions for meteor observations.

Keywords

Meteor simulation, Detection algorithm, On-board processing, Earth observation

¹ Small Satellite Student Society KSat e.V. Stuttgart, Germany, st166207@stud.uni-stuttgart.de

² Small Satellite Student Society KSat e.V. Stuttgart, Germany

³ Institute of Space Systems IRS Stuttgart, Germany

Nomenclature

d_{TV}	<i>selectable distance between camera and screen</i>
I_{curr}	<i>current pixel brightness</i>
I_{max}	<i>maximum pixel brightness</i>
S_{screen}	<i>screen size</i>
$Screen_{res}$	<i>screen resolution</i>
$Sensor_{res}$	<i>sensor resolution</i>
Z_{total}	<i>total frames for visible meteor</i>
Z	<i>current frame number</i>

Acronyms/Abbreviations

ArtMESS	<i>Artificial Meteorvideo Simulation Software</i>
MeSHCam	<i>Meteor, Star and Horizon tracking Camera</i>
PLOC	<i>Payload On-board Computer</i>
PERC	<i>Planetary Exploration Research Center</i>
SOURCE	<i>Stuttgart Operated University Research CubeSat for Evaluation and Education</i>

1. Introduction

This paper describes the working principle and results of developing a meteor simulation named Artificial Meteorvideo Simulation Software (ArtMESS). The task of ArtMESS is to create videos, showing meteors entering Earth's atmosphere as seen from orbit. Those videos are needed to test space-based meteor detection algorithms. ArtMESS can generate a variety of different, realistic meteors which allows systematic testing. Its use case include testing and optimizing a detection algorithm for the Stuttgart Operated University Research CubeSat for Evaluation and Education (SOURCE) satellite on ground using realistic data. The SOURCE satellite is a 3U+ CubeSat developed by KSat e.V. and the Institute of Space Systems at the University of Stuttgart and is part of ESA's *Fly your Satellite!* program [1]. Its mission objectives are education, technology demonstrations, re-entry science and meteor observation during eclipse. A visual camera is used for meteor observation. The observation generates a high amount of image data. This leads to the necessity of onboard processing since the downlink capacity is constrained. Due to the limited processing power and the absence of suitable detection algorithms, one must be developed and tested on ground. The camera used onboard the satellite is called Meteor, Star and

Horizon tracking Camera (MeSHCam) (see Table 1).

Table 1. Details of the MeSHCam

Parameter	Value
Camera	GenieNano M1920
Sensor resolution	1936 px × 1216 px
Pixel size	5.86 μm
Lens	Schneider-Kreuznach Cinegon
Focal length	12 mm
Field of view	31° × 48°

For the testing campaign and algorithm development a testbed is setup and calibrated in the IRS cleanroom. The testbed consists of the MeSHCam that is imaging a TV screen showing videos of meteors as seen from orbit. These images are saved and can be processed later or live using the PLOC. The algorithm then returns the positions and frames of the detected meteors.

2. Meteor Simulation ArtMESS

The meteor simulation is a Python based program. Fundamentally, it takes provided background videos, for example captured videos from the ISS by the Planetary Exploration Research Center (PERC) [2] and adds artificial meteors to provide suitable test data for an onboard detection algorithm. The following chapters will go into more depth on the three main parts of the program which are the used backgrounds, the meteor image generation itself and the implementation of several distortions (e.g., camera noise) that can occur observing meteors from a satellite.

2.1. General Principle

ArtMESS reads in a selected background video and saves each video as an individual image. Next, the meteor images are generated. Therefore, the number of images from the background video is counted. For each background image a mask image containing the meteor is generated, all other pixel values are set to 0. The mask images are combined into a three-dimensional array. Axes one and two of the array represent one mask image with the dimensions of the video (e.g., 1920 × 1080). The meteor pixel brightness is set according to adjustable parameters. The third axis represents the temporal progression. The length of the third axis is set by the number of background frames. Afterwards, the array is also divided into frames and

saved as individual images. These mask images are then further adjusted for distortions like camera noise. In a final step, both the initial background images and the mask images are put together into joint image frames and in the end into the final test video. All parameters set by the user or chosen randomly are written to a csv-file along with the assigned video name to allow for later review. This also includes the frame number and position in the frame of the implemented meteor. Thus, automated evaluation is enabled by comparing the meteor positions detected by the detection algorithm and the actual positions of the meteors. ArtMESS is also capable of operating in three different modes. In the first mode a single video is generated, the second one generates a whole test set which varies meteor parameters for systematic testing. Finally, the third mode allows to read parameters of a previous run and change some parameters (e.g., implementing the same meteor but using a different background).

2.2. Backgrounds

The used backgrounds can be divided into three categories: A black background, self-generated background videos from NASA's Black Marble [3] and ISS videos from PERC [2].

The black background is used to create baseline videos for testing. They can be used to test the algorithms general capability of detecting meteors. The self-generated background videos are used to test the algorithms' ability to distinguish city lights from meteors. The background videos are based on the black marble images and are generated using an additional small script using the sliding window method. For this method sections of NASA's Black Marble image [3], which shows the Earth at night, are used: A sliding window moves across the image plane, generating a series of frames that are put together in a video. This imitates a satellite's movement while observing Earth. This is possible due to the large resolutions of the black marble images. Finally, ISS videos provided by PERC [2] are used. These offer a lot of variety for example lightning, clouds, real meteors, or image errors like noise.

2.3. Meteor image generation

The detection algorithm determines areas that are not moving in the main direction caused by satellite movement. Hence the following parameters can influence the result of a successful detection (see [4] and [5] for more details on the algorithm).

Brightness: Brightness is an important factor not only influenced by the absolute brightness

but also by the light curve itself. The light curve of a meteor represents the variation in brightness as it burns up in the atmosphere. For a realistic result it is necessary to consider this property of a meteor. Therefore, a brightness function is used, which gives the current pixel brightness (I_{curr}) of the meteor depending on the duration of the meteor (Z_{total}) and its maximum brightness (I_{max}). The currently implemented functions is orientated on actual light curves measured [3]. It is implemented as shown in Eq. 1, where z is the current time step.

$$I_{curr} = I_{max} - \frac{4}{z_{total}^2 * I_{max}} * \left(\frac{z - z_{total}}{2} \right)^2 \quad (1)$$

For a more realistic depiction in the final video, each pixel brightness is multiplied by a random factor between 0.8 and 1.2. This simulates the burn up of the meteor in the atmosphere and considers the fact, that the meteor has not the same brightness of its spatial extension.

Speed and angle: Furthermore, speed and angle play a key role in the simulation. A similar meteor and satellite movement makes it harder for the algorithm to detect the meteor. Suitable and realistic values for speed have been found to be 1.5 to 7 pixels of meteor movement between two frames. A meteor is considered easy to detect when the angle between satellite and meteor movement is $180^\circ \pm 45^\circ$ while $0^\circ \pm 45^\circ$ are the hardest to detect. In this case the meteor moves almost in the same direction as the satellite.

Trail: The final parameter is the length of the meteor trail. It is used to further diversify the available test cases. The longer the trail the easier the algorithm recognizes the meteors due to their larger size. Values between 100 pixels to 500 pixels are good and realistic values.

It is important that the motion of the meteor is realistic with respect to background. Therefore, an optical flow is calculated for the background videos and is used to correct the movement of the meteor between frames.

2.4. Distortions

During space-based meteor observation, a variety of distortions can occur. The algorithm needs to deal with those. Radiation and camera noise can complicate detection and are therefore considered in the simulation. Implemented are e.g., gaussian noise, salt and pepper noise or hot pixel to name a few. The noise is layered over the final video frames, as one of the last steps.

Additionally, to the noise, the algorithm also needs to deal with limitations of the attitude control system: The satellite may rotate around the optical axis during observation. This effects the algorithm and needs to be considered during testing and development. Thus, the ArtMESS allows to generate videos with a settable rotation. It is implemented by using a bigger sliding window on NASA's Black Marble images [3]. This results in a larger background video with higher aspect ratio. After generating the meteor images and combining them with the background, the final video images are cut out and rotated by a few pixels each frame. This results in a rotation around the optical axis.

3. Testbed setup

As mentioned above, the testbed is used to display and image the generated videos using the meteor detection camera MeSHCam. The images taken by the camera are either processed with the meteor detection algorithm in real time or stored for later processing. This allows to test the camera, the control software as well as to optimize the algorithm by using realistic data: The images are acquired with the same camera and significant camera settings effecting the algorithm as planned for the orbital observations. Before describing the required calibration, a short overview of the testbed design is given.

3.1. Testbed Design

The testbed consists of a large OLED screen mounted on an aluminum frame. The camera is mounted on an optical rail to allow horizontal and vertical adjustments as well as providing a fixed and defined position.

When conducting measurements, the complete setup is covered in blackout curtain to prevent stray light from entering the setup as you can see in Figure 1.

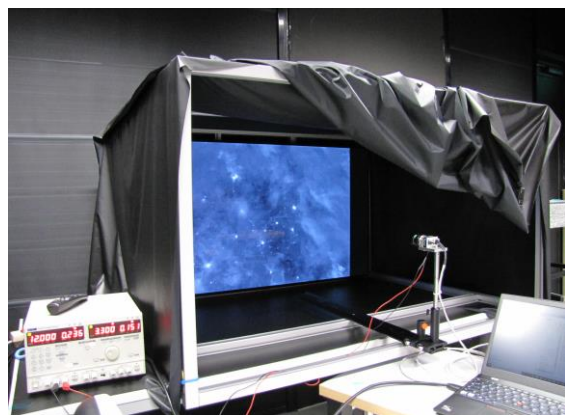


Figure 1: The testbed setup during algorithm testing in the clean room with MeSHCam and a generated video displayed.

3.2. Optical design

An OLED screen is chosen as the basis for the testbed, since pretests showed that screens using a backlight (e.g., LCD) cannot be used. This is due to the fact, that the backlight emits light even for complete dark parts of a video (high black level). The light is imaged by the camera and results in unrealistic images. Moreover, the issue is worsened due to the uneven distribution of the light generated by the screen. An OLED screen uses individual LEDs for each pixel, thus in black parts of the video the LEDs are turned off resulting in a superior black level.

Two aspects are considered when designing the testbed:

- 1) The size of the screen must be large enough to cover the complete FOV of the camera at a useful distance. A useful distance is defined as large enough that the camera still can be focused (614mm for the used lens) but small enough to keep the setup at reasonable dimensions (not larger than 1.3m).
- 2) At the chosen camera distance, the screen resolution should be large enough. This means, one pixel of the camera image should image more than one screen pixel.

The screen size can be derived from basic geometry using the FOV and selectable distance between camera and screen (d_{TV}).

$$S_{screen,v} = \tan(FOV_v/2) * 2 * d_{TV} \quad (2)$$

The vertical FOV is used since it reaches the edge of the screen first, therefore limiting the distance. This is due to the higher screen ratio (16:9=1.77) compared to the image sensor ratio (1936px/1216px=1.6). Using the screen ratio and calculated vertical screen size, the screen diagonal can be calculated.

At the time of the testbed design, only one affordable OLED screen (LG OLED55B8LLA) was available with a diagonal of 55 inch. This means, the maximum distance is about 1.2m, before the FOV expands over the screen. The actual distance used is 1m to keep the setup smaller.

The second aspect, number of screen pixels imaged on one camera pixel, is calculated with the resolution of the screen (3840px*2160px) and camera (1936px*1216px). At the given distance of 1m the size of the screen imaged is determined by using Eq. 3. Next, the number of screen pixels imaged is calculated. Finally, the pixel density can be determined using the number of screen pixels imaged and camera resolution.

$$\text{Pixel Density} = \frac{\text{Screen}_{res,x}}{\text{Sensor}_{res,x}} \quad (3)$$

For a distance of 1m, the pixel density is sufficient with 1.46 screen pixels per camera pixel.

3.3. Testbed Calibration

The testbed can only represent a proper environment if the settings of the hardware and the input images are adjusted accordingly. This will ensure that the images the MeSHCam outputs are resembling the view of an Earth observing satellite to test the detection algorithm. The calibration mainly counters the specific brightness curve of the TV screen: Instead of a linear relation between input pixel value and output pixel value, a quadratic relation was observed as can be seen in Figure 2. This is an effect of the screen because the output pixel values of MeSHCam are scaling linear with the received irradiance.

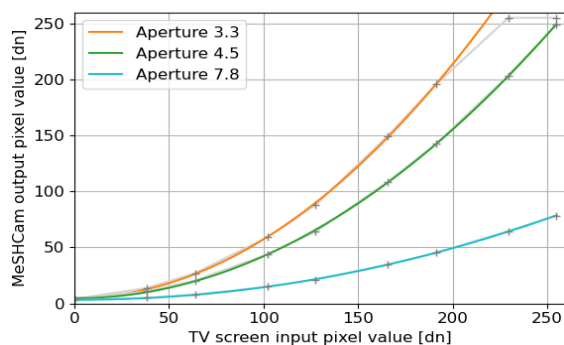


Figure 2: Output pixel of MeSHCam plotted over input pixel to the TV screen at different apertures.

3.3.1. Relative Brightness Calibration

To assure consistency between tests, specific TV screen settings were fixed. The settings were chosen in a way that leaves little to no room for the screen to modify the input. For example, OLED-Light and Dynamic Contrast were shut off. The brightness and contrast settings were selected because their specific combination resulted in the least curved input pixel value vs output brightness relation.

The next step of the brightness calibration is to determine a specific lens aperture. To use the screen to the maximum extent, a displayed pixel of maximum brightness should produce a pixel close to maximum brightness in the image the camera outputs. To achieve this, an array of 8 images with different luminosities (0 to 255) was shown on the screen and imaged by MeSHCam with different apertures in the range from 1.4 to 11. The results provided the best fitting aperture as well as enough data points to calculate the regression curve of all possible pixel values shown in Figure 2. In our case, aperture 4.5 was

chosen because it met the earlier specified criteria best. The regression curve can now be used to predict the output pixel values in the camera given the input pixel values of the screen. Through reversing the curve, it is possible to determine the input needed for a specific output. In the meteor simulation, a Python function creates a lookup table and then swaps all pixel values in an input frame. If a modified frame is imaged by the MeSHCam, the brightness in the resulting image will be almost identical compared to the original input frame as can be seen in Figure 3.

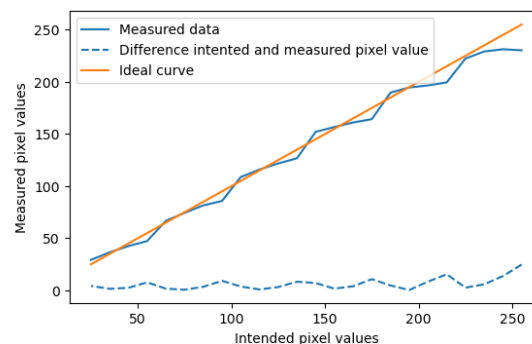


Figure 3: Comparison of pixel values after the calibration's adjustment.

3.3.2. Geometric Calibration

Automated test evaluation using the known meteor position in the video and the detected meteor position determined by the algorithm improves the development speed of the algorithm significantly. It is achieved by mapping pixel positions on the screen to pixel coordinates in the camera image. This requires a fixed camera: The camera must be centred and rotationally aligned with the screen. In order to take into account small inaccuracies in camera alignment between individual tests, a chequerboard-style image is imaged by the MeSHCam. The corners of the squares can be automatically read out, thus creating a rough map. Finally, interpolation between coordinates allows the mapping of all coordinates from the taken image back to the original frame. If the algorithm detects a meteor at a specific position in the image, the position can be mapped to coordinates in the original image. Since the meteor position in the original video is known from ArtMESS, it can be checked whether the meteor was identified correctly.

4. Results

The meteor simulation works and is used to develop and improve the detection algorithm.

4.1. Application of the simulation

In the scope of a master thesis, the simulation was used to develop and improve the algorithm

[4]. Therefore, a systematic set of test videos was generated. This test set contains videos with different backgrounds, each is used for a different purpose: The black background is used to evaluate the effect of algorithm parameters on the detectability of meteors with different properties (e.g., speed and brightness). The black marble backgrounds [2] are used to determine the effect of a moving background and adapt the parameters accordingly. Various videos taken from the ISS [2] are used to consider clouds and lightning.

All in all, the simulation was crucial for testing, improving and setting parameters for the algorithm. Furthermore, the export of the meteor position in the generated videos allows for an automatic evaluation of the results since the algorithm also exports the meteor position and time of detection. Using a Python script to automatically evaluate the detection performance significantly reduces the development time.

4.2. Comparison of artificial and real meteor

To validate the simulation and assure realistic data is generated, the videos and implemented meteors were compared to the meteor videos taken from by PERC from the ISS. As can be seen in Figure 4, the artificially generated meteor looks similar to a real meteor. Thus, the generated videos can be used for further development.

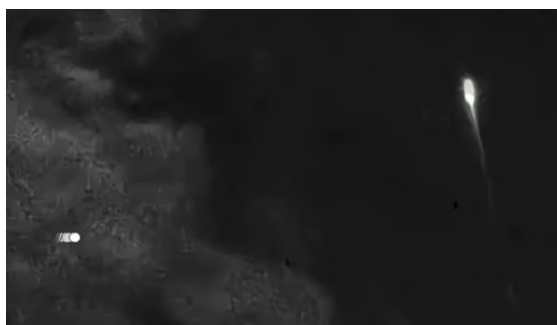


Figure 4: This image shows two meteors. On the left you can see an artificial meteor and on the right is a real meteor for comparison. [3]

5. Discussion / Conclusion

As outlined in the previous section, ArtMESS is a crucial tool in the development of the novel space-based onboard meteor detection algorithm for the SOURCE mission. Therefore, the development of the detection algorithm and testing it with ArtMESS is mission critical. ArtMESS will be used to further improve the detection algorithm and will also be further developed. For example, the formula for the light

curve can be improved and adapted to represent a more realistic meteor burn up. Since other space-based meteor observations missions are planned and each of them relies on a working detection algorithm, we decided to publish ArtMESS under an open-source license. ArtMESS will be published this year on the Institutes Software Repository⁴.

Acknowledgments

A special thanks goes to all students who have been and are working onwards a successful SOURCE mission. Additional thanks go to the academic personnel at the IRS and students of KSat who supported us with their time and expertise. We also want to thank ESA for their support with the *Fly your satellite!* program and all sponsors of SOURCE without their help the realization of such a project would not be possible.

References

- [1] A.Stier,R. Schweigert,D. Galla,M. Lengowski,S. Klinkner, Combination of Interdisciplinary Training in Space Technology with Project-Related Work through the CubeSat SOURCE, 3rd Symposium on Space Educational Activities, Leicester, United Kingdom, 2019
- [2] T Arai et al. "Meteor observation HDTV camera onboard the international space station". In: Lunar and Planetary Science Conference. 1777. 2014, p. 1610
- [3] Steven D. Miller et al. "Suomi satellite brings to light a unique frontier of nighttime environmental sensing capabilities". In: Proceedings of the National Academy of Sciences 109.39 (2012), pp. 15706–15711. issn: 0027-8424. doi: 0.1073/pnas.1207034109
- [4] BOROVIČKA, J.; SPURNÝ, P.; KOTEN, P. Atmospheric deceleration and light curves of Draconid meteors and implications for the structure of cometary dust. *Astronomy & Astrophysics*, 2007, 473. Jg., No. 2, p. 661-672.
- [5] Julia Zink. "Test and optimization of an onboard meteor detection algorithm for the CubeSat SOURCE (IRS-21-S-066)". English. Master Thesis. Stuttgart, 2021
- [6] Jona Petri "Satellite formation and instrument design for autonomous meteor detection", PhD thesis, in review, 2022

⁴ <https://egit.irs.uni-stuttgart.de>